Algorithms in Genome Research
Winter 2006/2007

Exercises

Number 8, Discussion: 2007 January 19

1. Word puzzling:

   - How many different words can be built by using all the letters of the word ALGORITHMUS exactly once? Compute the actual value.

   - How many different words can be built by using all the letters of the word ABRACADABRA exactly once? Note that all words need to have the same length, and must use the letters the specified number of times: For ABA, there are three such words, AAB, ABA, BAA.

   - Try and find a general formula. Hint: For ALGORITHMUS, the formula depends only on the length of the word, but not for ABRACADABRA - what else does it depend on?

2. Suppose that we do not know the order of characters in a string: For example, the strings AACCC, ACACC, ..., CCCAA are indistinguishable to us. We call such "strings without order" *compomers* (denoted A_2C_3 for our example). The *length* of a compomer is the length of the corresponding string (5 in our example).

   - Let $\Sigma = \{A, C, G, T\}$ be our alphabet, then there exist 4 compomers of length 1 (A_1, C_1, G_1, T_1) and 10 compomers of length 2 ($A_2, A_1C_1, C_2, A_1G_1, C_1G_1, G_2, A_1T_1, C_1T_1, G_1T_1, T_2$). How many compomers exist of length 3 and 4?

   - Derive a general formula for the number of compomers of length n over an arbitrary alphabet Σ.

3. Given a list of peaks from a Tandem Mass Spectrum (MS/MS) for peptide de-novo sequencing, one important obstacle for recovering the peptide sequence is to assign peaks to the main ion types b and y (prefix and suffix strings of the peptide sequence). If we know that only b-ions are present in the spectrum, then recovering the sequence becomes simple. Describe an algorithm to do so: Input is an ordered list of masses $m_1, \ldots, m_n$.

4. We modify the above problem such that there are "noise peaks" (of unknown origin) in the mass spectrum. Describe an algorithm that finds a peptide sequence maximizing the number of explained peaks. The algorithm should run in $O(n|\Sigma|)$ time. Hint: use one-dimensional dynamic programming.